PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 15/00, A61N 1/30	A1	(11) International Publication Number: WO 94/23375 (43) International Publication Date: 13 October 1994 (13.10.94)

(21) International Application Number: PCT/US94/02919

(22) International Filing Date: 17 March 1994 (17.03.94)

(30) Priority Data:
08/040,633 31 March 1993 (31.03.93) US

(71) Applicant: LUMA CORPORATION [US/US]; 311 Courthouse Road, Princeton, WV 24740 (US).

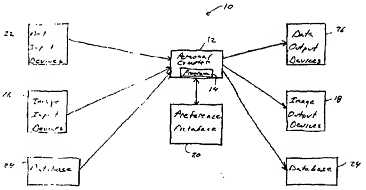
(72) Inventor: BRANSON, Philip, J.; Rt. Six, Box 578, Ingleside Road, Princeton, WV 24740 (US).

(74) Agent: DEVLIN, Peter, J.; Fish & Richardson, 225 Franklin Street, Boston, MA 02110 (US).

(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: MANAGING INFORMATION IN AN ENDOSCOPY SYSTEM



(57) Abstract

A system (10) for acquiring images during a medical procedure and using the acquired images includes a storage device (24) for storing, for each one of a plurality of users of the system (10), information that indicates one or more processing operations to be performed on images obtained by an input device (22). A system processor (12) responds to an identity of the user who is currently using the system (10) by performing processing operations on the obtained images and applying the images to an output device (18) based on the stored information that corresponds to the current user.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgystan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SI	Slovenia
CI	Côte d'Ivoire	KZ	Kazakhstan	SK	Slovakia
CM	Cameroon	LI	Liechtenstein	SN	Senegal
CN	China	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
CZ	Czech Republic	LV	Latvia	TJ	Tajikistan
DE	Germany	MC	Monaco	TT	Trinidad and Tobago
DK	Denmark	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	US	United States of America
FI	Finland	ML	Mali	UZ	Uzbekistan
FR	France	MN	Mongolia	VN	Viet Nam
GA	Gabon				

MANAGING INFORMATION IN AN ENDOSCOPY SYSTEM

Background of the Invention

This invention relates to managing information in an endoscopy system, and in particular to controlling the acquisition, processing, storage, and display of endoscopic images.

During an endoscopic surgical procedure, the surgeon typically uses a video camera (or other suitable device, such as a video arthroscope) to capture images of the surgical site. The images are generally applied to a display device (such as a television monitor) for observation. In some cases, the images are stored on video tape (by, e.g., a VCR) or are converted to digital files for storage in memory or on a disk drive. Different physicians often use the video equipment in different ways.

Summary of the Invention

In one general aspect of this invention, a system for acquiring images during a medical procedure and using the acquired images includes a storage device for storing, for each one of a plurality of users of the system, information that indicates one or more processing operations to be performed on images obtained by an input device, and a processor that responds to an identity the user who is currently using the system by performing processing operations on the obtained images and applying the images to an output device based on the stored information that corresponds to the current user.

Preferred embodiments include the following features.

The stored information also indicates, for each user, a configuration of the input device and a configuration of the output device. During operation, the input and output device configurations are

established in response to the identity of the current user. The configuration of the input device includes the format in which the device produces the image, and the configuration of the output device includes the format in which the device uses said image. As a result, the system can translate images obtained in one format (e.g., NTSC, RGB, Y-C, PAL, etc.) to another format for display by the output device, all according to the user-specified information in the storage device.

The information further indicates, for each user, a sequence of images that are to be obtained during the procedure. During operation, the current user is prompted to obtain the images in the sequence. This allows the user to follow pre-established "scripts" (stored as directed by the user) to perform tours of the surgical site according to the user's preference.

The storage further stores the information for a plurality of surgical procedures. The processor responds to an identification of the surgical procedure currently being performed (as entered by the user) by performing the processing operations based on the information that corresponds to the current surgical procedure. Thus, each user may specify, for example, different tours for the various surgical procedures that he or she performs.

The system includes a plurality of input devices and multiple output devices. The storage device storing the information for each input device and each output device. During operation, the processor selectively processes images from the input devices based on the identity of the current user and the stored information. Likewise, the processor selectively applies the processed images to the output devices based on the identity of the current user and the stored information.

The storage device stores the information as a plurality of linked records. A first one of the records

identifies a type of the information, and a second one of the records contains data associated with that type of information.

As described in more detail in the description of preferred embodiments, the invention provides numerous other features and offers several advantages. Among them are:

1. a standardized platform for endoscopic video systems that allows more rapid evaluation and improvement of hardware and functionality.

2. simple acquisition and storage of still and live images from the operating room or endoscopy suite.

3. digital enhancement of acquired images for presentation in printed and film formats.

4. annotation of acquired images with text and graphics.

5. formatting and providing composites of multiple images, images and stock images, or drawings in a manner that provides spatial, logical or other context that improves the communicating value of the acquired images.

6. a stand alone database application that allows individual doctors to maintain and retrieve images and text data.

7. an interface that allows existing office or hospital databases to manage image and text information relating to endoscopic applications.

8. allowing existing endoscopic database and endoscopy suite management systems (such as Siscope and others) to integrate imaging into their applications.

9. improving the reliability and dependability of endoscopic image and text data by facilitating simple and reliable systems for integration of the data with physician and facility work flows and hardware constraints.

In addition, the invention implements a modular architecture that allows improvement by upgrade instead of replacement. Regulatory requirements can often be satisfied by a modification to an existing filing, as opposed to a de novo filing.

Other features and advantages of the invention will become apparent from the following detailed description, and from the claims.

Brief Description of the Drawing

Fig. 1 is a block diagram of a system for managing endoscopy information according to the present invention.

Fig. 2 shows one possible configuration of the system of Fig. 1.

Figs. 3 and 4 shows details of a subsystem used in Fig. 2.

Fig. 5 shows two subsystems of Fig. 2 connected in parallel for so-called "picture-in-picture" display of images.

Fig. 6 illustrates a procedure for capturing and storing images in the system of Fig. 1.

Fig. 7 shows a "tour" procedure for capturing images in the system of Fig. 1 according to a predefined script.

Fig. 8 is useful in understanding how the system of Fig. 1 formats printed outputs of images.

Fig. 9 illustrates a procedure for annotating images obtained by the system of Fig. 1 with text and graphics.

Fig. 10 is useful in understanding a character recognition feature of the system of Fig. 1.

Fig. 11 shows a procedure implemented by the system of Fig. 1 to convert the video format of an input image to a different video format for use by an output device.

Fig. 12 is useful in understanding the operation of a remote control feature of the system of Fig. 1.

Fig. 13 shows a procedure performed by the system of Fig. 1 for producing slides from the acquired images.

Fig. 14 shows the device handling architecture of the system of Fig. 1.

Fig. 15 illustrates the use of universal connectivity devices (that use, e.g., the PCMCIA standard) in the system of Fig. 1.

Fig. 16 is useful in understanding the operation of a preference database of the system of Fig. 1.

Figs. 17 and 18 illustrate the storage of acquired images in an image database of the system of Fig. 1.

Fig. 19 is a flow chart that shows the operation of the system of Fig. 1 from the standpoint of the user.

Fig. 20 shows modules of a software program that controls the overall operation of the system of Fig. 1.

Fig. 21 is useful in understanding the structure and organization of the databases used in the system of Fig. 1.

Description of the Preferred Embodiments

Referring to Fig. 1, endoscopic procedure management system 10 employs a personal computer 12 which executes a stored program 14 to configure and manage all devices used by a physician during endoscopy. In particular, personal computer 12 receives images generated by one or more image input devices 16, processes the images according to the preferences of the physician performing the endoscopic procedure, and transmits the processed images to one or more image output devices 18 for display or storage.

Personal computer 12 obtains information concerning the physician's preferences about the processing, display, and storage of the images from a preference database 20. As described in more detail

below, preference database 20 is organized by physician -- that is, the preferences of each physician recognized by system 10 as to the image processing and the configuration and operation of image input devices 16 and image output devices 18 are stored in preference database 20 according to the identity of the physician and according to the various procedures performed by the physician. As a result, when a physician logs onto system 10 (as described below) and identifies the endoscopic procedure that he or she is to perform, host computer 12 configures image input devices 16 and image output devices 18 and controls the processing of the obtained images according to the information stored in preference database 14.

A wide variety of image input devices 16 can be used with endoscopic procedure management system 10. For example, image input devices 16 include video endoscopes, cameras, video cassette recorders (VCRs), X-ray scanners (which convert X-ray films to digital files), digital X-ray acquisition apparatus (which convert X-ray photons to a digital bitstream or file), fluoroscopes, CT scanners, MRI scanners, ultrasound scanners, and other types of scanners (handheld or otherwise).

Likewise, several different types of image output devices 18 can be connected to system 10. Examples include cathode ray tube (CRT) displays, television monitors, LCD panels and screens, EMM memory, and disk memory. The displays may use the RS170 format (i.e., RGB, Y-C, NTSC, PAL, or the PAL equivalent of Y-C), or alternatively may be non-RS170 devices (interlaced and non-interlaced display devices). As described in detail below, video subsystems connected within personal computer 12 define communication protocols (e.g., horizontal and vertical synch, interlaced vs.

noninterlaced display, scan speed, and spatial and color resolution) to image input and output devices 16, 18,

In addition, personal computer 12 also receives data and commands from one or more data input devices 22. Data input devices 22 include computing devices such as laptop and palmtop computers, as well as pen-based command systems. Data input devices 22 also include programmable input devices (such as infrared remote control units), and pointing devices such as a keyboard, mouse, trackball, airmouse, light pen, touch screen, and tablet. All of these devices allow the physician (or another used located, e.g., in the sterile field) to manage the operation of system 10, as described in detail below.

Personal computer 12 can also be connected (either directly or via a network or modem) to one or more databases 24 to receive patient and other information. Examples of databases 24 include hospital information systems and local databases located in the physicians' offices.

Numerous data output devices 26 are also connected to personal computer 12 for receiving data concerning the operation of system 10 (e.g., reports). Personal computer 12 also sends data to one or more of databases 24 for storage. Examples of data output devices 26 include laptop, palmtop, and pen-based computer systems.

Referring also to Fig. 2, one possible configuration of system 10 is shown. Personal computer 12 includes a host processor 30 and a series of subsystems or adapters 34a-34p (hereinafter generally referred to as 34) each of which is connected to host processor by a bus 36. (If the configuration of personal computer 12 so dictates, bus 36 is replaced by a backplane.) If needed, one or more subsystems 34 can

also be connected to host processor or to each other via a local bus (not shown).

Host processor 30 provides an industry standard platform which supplies a common power supply, a computing function (CPU 38), memory and storage devices 40, mechanical support and cooling functions 42. Subsystems 34 gain access to any or all of these services via bus 36 in a motherboard implementation (or the backplane in a backplane implementation). Other systems outside the physical housing of personal computer 12 can gain access via subsystems 34 that externalize these functions from personal computer 12. Host processor 30 is a 486-66 based machine; alternatively, host processor 30 may be implemented as a workstation. Standard ISA bus architecture is used, but EISA architecture or others can be selected instead. In the motherboard configuration, the memory nad CPU and expansion slots are on a single board. In the backplane configuration, power and communication occur via the backplane, and CPU 38 and memory 40 plug into the backplane like an adapter.

Subsystems 34 and host processor 30 communicate in a bidirectional fashion over bus 36. Each issues and receives commands, status and other information. Host processor software control 14 integrates and regulates the flow of information, and hence the operation of subsystems 34 and the input/output devices. Communications that conform to platform standards can be handled by bus 36. Specialized or nonstandard communication can be handled by local buses, like a local video bus (not shown). A local bus provides specialized communication of limited scope. Overall function of the local bus can be governed by communication that occurs over bus 36.

The communication afforded by bus 36 allows for software control of the specific functions of system 10. For example, preference database 20 can contain information regarding video gain and offset, image format preferences, and text data required for display of images obtained during endoscopy. Multiple settings or preferences can be accommodated and tailored to the (typically different) specifications of each physician. System configuration and real time operation are monitored by software defined logic to allow for dynamic optimization of all controlled functions. Decisions can be made based on operator input, like specification of procedure or surgeon. Decisions can also be made based on system generated input, like video level, color composition, or most recent operator interactions. Preferences can be modified by the operator, or dynamically by the system.

As discussed below, multiple control devices are supported. An infra-red, programmable handheld remote control 48 provides operator interaction from a sterile field. Hard wired switches connected by wires to the operative field can also be used. The system also provides for multiple inputs from touchscreens, mouse, light pens and other relative and absolute pointing devices 56.

The system is structured to be platform independent. The software provided for the system is organized so that portions of the code that are hardware dependent can be easily replaced. As CPU and adapter subsystems are improved or changed, a means is provided to allow simple integration of the new devices. Input and output devices and specifications are stored in preference database 20, and control device specific outputs. For example, 24 bit color images can be

processed to display on a 64 color gray scale device based on the database preference.

As also discussed below, the software system include heuristic systems that allow multiple complex functions to be controlled automatically or with minimal and simple operator input.

The host/subsystem architecture allows for both standardized communication therebetween and the sharing of resources (such as memory and processing functions). Subsystems 34 identify themselves to host processor 30, send queries to and respond to queries from host processor 30, and provide signal and control outputs to host processor 30.

Communication occurs over bus 36 between host processor 30 and subsystems 34 and between the subsystems 34 themselves. As can be appreciated from Fig. 2, inputs can be applied to personal computer 12 either through the subsystems (e.g., subsystems 34a-34g) or directly to host processor 30. Likewise, output signals and data are produced by both host processor 30 and the subsystems (e.g., subsystems 34j-34p).

Fig. 2 illustrates several different types of subsystems 34. Camera subsystems 34a, 34b provide host interfaces for a camera 44 and a video endoscope 46. Remote control adapter 34c interfaces a remote control unit 48 (infrared or otherwise) to host processor 30. The physician can control system 10 using voice commands that are relayed from voice pickup 50 to host processor 30 by voice control adapter 34d. A fluid/gas monitor 34e relays signals from a fluid gas sensor 52 in the sterile field to host 30. Laptop, palmtop, and other computing devices 54, and pointing devices 56 communicate with host processor 30 via suitable adapters 34f. PCMCIA devices (discussed below) are interfaced with host processor by PCMCIA adapters 34g. Other input devices 58 (such as a

keyboard and pointing devices such as a mouse) are connected directly to bus 36.

Video subsystems 34j, 34k provide interfaces with a CRT display 60 and a television monitor 62. The level of light provided in the surgical field by an illuminator 64 is controlled by light source subsystem 34l. A VCR subsystem 34m supplies image signals to a VCR 66. The operation of devices used in the surgical field, such as fluid supply 68 and one or more medical devices 70 (e.g., surgical tools such as shavers, abraders, end cutters, etc.) are controlled via fluid management subsystem 34n and medical device subsystem 34o, respectively. Other output devices, such as printers and film recorders 72 are driven by host processor 30 via one or more PCMCIA adapters 34p, while some output devices 74 can be connected directly to bus 36.

Preference database 20 can also be connected to host processor 30 via a PCMCIA adapter 34i. A PCMCIA adapter 34h is used to connect host processor 30 to units 76 that provide other services. Examples of other units 76 include storage devices, e.g., disk drives, memories (such as so-called "flash memories"), disk drives, and network adapters (for wire, wireless, RF, or optical networks). But any device that conforms to the PCMCIA connectivity standard (or any other suitable standard, such as JEIDA) can be connected to personal computer 12 in this fashion. In addition, any suitable port of host processor 30 (e.g., serial, parallel, SCSI, etc.) can be used to provide I/O capability.

Referring to Figs. 3 and 4, details of video subsystem 34j are shown. (It will be understood that video subsystem 34k is identical to subsystem 34j.) Video subsystem 34j resides on full length PC card, and contains the necessary processors and memory to allow real time gain and offset control, and matching of

multiple inputs to multiple output timing specifications for display, pan, zoom, video overlay and digital signal processing.

The multiple functions of video subsystem 34j include:

1. Encoding and decoding RGB, NTSC, Y-C and PAL.
2. Synchronization functions: acquisition synch. display synch, pixel clock control and synch conversion.
3. Digital signal processing (DSP).
4. Image manipulation operations such as zooming and panning.
5. Drawing graphics such as boxes and lines.
6. Control of video memory (i.e., in, out).
7. Controlling aspect ratio.
8. Select between display of: (a) the image that currently resides in the frame buffer, or (b) the live image.
9. Definition of input and display device parameters.
10. Acquisition and display look up table controls.
11. Acquisition and display weighting, adjustment.
12. Color extraction and object recognition functions.
13. Masking functions.
14. Video display/overlay keying.
15. Convolution.
16. Management of programmable events.
17. Mathematical operations on video signals.
18. Inter-image operations and manipulations.
19. Histogram extractions from images.
20. Operations on defined areas of interest.
21. Morphologic manipulation of image information.

22. Extraction of a profile of pixels for a defined sample range.
23. Control of pixel flow between host and subsystem memory.
24. Extraction of the run-length of an image.
25. Controlling the transfer of host and adapter memory buffer lines.
26. Controlling the range and offset of analog to digital converter.
27. Write and position text.

Many of these functions may be reproduced in software for still image processing and real time processing in virtual memory of host processor 30.

As shown in Figs. 3 and 4, video subsystem 34j receives an image 80 (to be displayed, e.g., on CRT display 60) in any suitable format (such as RGB, NTSC, Y-C, PAL, or PAL Y-C). The image is applied through an analog signal conditioner 82 and a range and offset controller 84 to an analog to digital converter 86. The digitized image is applied to a synchronization module 88, which also receives as inputs a camera synch signal 90 (and possibly other external triggers 92). Synchronization module 88 can also generate synch signals for one or more input devices (such as cameras). The synchronized image signal 94 produced by module 88 is passed to graphics processor 96.

Graphics processor 96 performs the graphical functions described above (several of which are listed in box 98) on image signal 94. The image produced by graphics processor 96 is sent to frame buffer memory 100 for subsequent display. Graphics processor 96 also returns information about the live, processed image to host CPU 38 via bus 36 (or via a local bus within personal computer 12). CPU 38 responds to this information, and to the display preferences (from

preference database 20) of the particular physician who is using system 10, by dynamically adjusting one or more display parameters of the live image.

Examples of the processing performed by CPU 38 on the image include masking 102, keying 104, VGA-, video source-, or CPU generated-overlaying 106, and output leveling 108. In addition, CPU 38 adjusts 110 (where required by the preferences from database 20) such display parameters as display synch, pixel rate, and frame control. The resulting image information (consisting, e.g., of BLUE LUT, GREEN LUT, RED LUT, and SYNCH) provides the output 112 of video subsystem 34j.

The zooming function performed by graphics processor 96 allows real time images or still images to be magnified by a user specified value. The zoom occurs concentrically about a point specified by the user or preference database 20, and is usually the central point of the image. Image processing is done to allow highest quality view of the magnified image. This is either a specific hardware routine or can be performed in software in a virtual frame buffer.

In addition, execution of the zoom function adjusts an electrical potential output (not separately shown) that allows any external device to interpret the potential and adjust light intensity so that the light level is appropriate to the field of view. The electrical potential is related to the luminance or brightness of the reduced field of view caused by the zoom, as opposed to the entire incoming video signal. This allows appropriate light level for the area being visualized.

If an image is saved (i.e., "captured" as discussed below) during zooming, a tag is placed in the image file. The tag alerts an output device that uses the image file to display or print the image as zoomed,

i.e., in the same manner in which the image was displayed to the user at the time of capture.

In the non-zoom, or 1:1 mode, area of interest processing can be used to generate a light source control potential that is weighted for a defined area which is a subset of the entire viewing area. This area of interest can be dynamically changed by an algorithm in the software.

System 10 also provides for manual, user prompted control of gain and offset, multiple values for gain and offset specified in preference database 20, and a real time process for automatic context and value sensitive means for control of gain and offset. Automatic adjustment is accomplished by mathematic processing of signal in areas of interest on the live or incoming video.

The architecture of video subsystem 34j allows images to be displayed in a variety of formats according to the preferences of the physicians that use system 10. For example, referring also to Fig. 5, a pair of video subsystems 34j, 34k are wire-ored together to produce a "picture-in-picture" output for display. Both images are applied to each video subsystem 34j, 34k. Each subsystem 34j, 34k processes one image and simply monitors the other image for synchronization purposes. For example, video subsystem 34j processes input 1 and monitors input 2, leaving it to video subsystem 34k to process input 2 (and monitor input 1). More than two inputs can be used, if desired.

The image inputs may be either synchronous or asynchronous. That is, video signal synchronization can occur at the inputs to subsystems 34j, 34k via a system generated synch signal, or at the level of subsystems 34j , 34k in display management by CPU 38. Host CPU 38 and program 14 control the output video, the asynchronous

video signals to a single output display. The wired-or connection provides a current drive that adds the two image signals together. The sizing, location, and priorities of the two images are selected, switched, and defined based on the preferences set forth in database 20 and the sterile field controls (e.g., remote control 48, Fig. 2) operated by the physician or other user. Each signal input can be processed controlled with the multiple functions discussed above. The CPU commands for handling the multiple images are in addition to the commands listed above, and are relayed to the subsystems 34j, 34k via bus 36.

Referring to Fig. 6, another function performed by video subsystem 34j is capturing images from frame buffer 100 (Fig. 4) and storing the images in memory. Capture can occur by direct operator interactions (via, e.g., remote control 48 or a pointing device). A single button press can take a snapshot of the current information on the screen and transfer it to a memory media as a digital file. This system provides additional logic to allow toggle between "freeze" and "live," until the desired image is frozen. The image can then be saved to memory (by executing save command 120, so the surgeon can elect to save or discard any image.

The freeze and capture functions can be controlled by any data input device 22. The current implementation allows control by hardwired buttons on the camera or infra red handheld remote control. The review function 122 displays all images captured, either as a composite of many postage stamp size images, or as a sequence of full size images. This allows the operator to be sure adequate images have been captured.

Fig. 7 illustrates a "tour" function performed by system 10. Preference database 20 contains, for each physician who uses system 10, one or more "scripts" 130

for designated captures of images. When the tour function is initiated 132, system 10 prompts the physician 134 to capture images by anatomic site, pathology or other designation. Prompting is performed via a graphical object or a text clue imposed on the displayed image that guides the operator to the proper position for capture. The physician has the opportunity to skip an image or acquire additional images 136 during the tour. Upon the completion 138 of the script, system 10 has captured and identified a series of images 140 specified in the script.

Multiple, i.e., pre-op and post-op tours can be designated 142. Pre-op and post-op tours advantageously show pathology and post endoscopic procedure results or status. If multiple tours are specified, system 10 selects 144 the scripts for the tours from preference database 20 in a logical sequence that corresponds to the natural flow of the surgical procedure. The format of the printed output may identify the sites as pre- and post-intervention.

Referring also to Fig. 8, the tour function also includes automatic annotation of images obtained in each tour with predefined text or graphical information 150. The tour can also specify composition of printed or other output, combining acquired images and/or stock images or diagrams that lend clarity and context to the images acquired (step 152). Page formatting allows multiple images in multiple sizes to appear on a single page or on multiple pages.

Page layouts can be defined and modified based on a taxonomy defined when the user defines the functions of tour. Currently, the page description language is post-script level 2, but other interpreters for conversion to other file formats can bee added. Images captured in the tour mode will define how they will be displayed,

arranged and labelled. Additional drawings, diagrams or stock images can be pulled into the page layout defined. Once defined, the layout can be stored, or printed to hard copy or slides.

Images captured randomly (outside of a defined tour) in an unstructured capture 146 are formatted by a user defined algorithm, or a default format is used in the absence of user instructions to the contrary. This layout formats pages to conform to a predetermined optimum number of images and spatial arrangement based on the target printing or display device specified in the user preference (as set forth in preference database 20).

In a session where both tour and random images are acquired, a page layout for the prints will be generated consisting of a tour format plus a random format based on desired output device. The user can configure and modify the layout at any time. If no user selection is made, a default layout is used.

Referring to Fig. 9, system 10 also provides a simple technique for overlaying text or graphics on the acquired images (the technique can also be performed via a laptop or palmtop computer or other suitable computing device that is externally connected to personal computer 12). The resultant image is a composite of the annotation and overlay, so that the acquired image remains unchanged. The overlay text is anti-aliased for improved appearance. Anti-aliased text is used extensively throughout the interactions with the system.

The starting point for the annotation procedure is the image file 160. The operator (e.g., the physician) selects the text or graphical objects with which to annotate the image using one or more suitable data input devices 22 (Figs. 1 and 2) (step 162). The input device is also used to locate the text or graphical object on the image (step 164. Object attributes such as font,

color, and size, can be selected manually or may be supplied by preference database 20 (step 166). The information that specifies the size, location, and other attributes of the graphical object are stored as relative coordinates in an overlay (step 168), and thus do not affect the source image. Finally, the objects and text are resized as the image is scaled for display (step 169). Printed output of the image/overlay composite will appear as if the overlay were part of the image itself.

Annotation can be performed in a similar manner at both the image and page layout levels. A single image or page layout can possess multiple annotation overlays. Single or multiple overlays may be composited with the image. The overlay may also possess the attribute of translucency, ranging from obliteration of the underlying image to transparency in areas in which the overlay is not visible. Single or multiple overlays may be attached to an image at any time.

System 10 also displays system status and functions to the user in a format and style specified by the user. A CPU-directed video keyer (Fig. 4) adds the status overlay to the video signal displayed by the system, in much the same manner as discussed above. The overlay includes text information, graphics, or other images.

Fig. 10 illustrates several different options available to the operator for entering the annotation information. An important functionality is "pen based" computing using a commercially available pen-based hardware 54 (Fig. 1) as the input device, but similar functionality may be obtained with other input devices. The endoscopic image is displayed, and the input device (e.g., the pen-based system) is used to annotate, manipulate, or process the image as discussed above. Using the pen-based system, the operator actually prints

or writes on the screen, and system 10 translates the pen strokes into machine-readable characters. System 10 includes a search/shorthand capability to expand characters to an expected or defined word or most common "near hit."

The pen-based system also provides an extremely straightforward device for allowing the user to draw lines, squares, and circles, and for controlling area of interest processing and color.

As discussed above, annotations are stored as text and location so that when video formats are changed, or images are scaled to larger or smaller sizes, that aspect ratio, location and contrast are maintained.

As discussed in more detail below, system software 14 (Fig. 1) includes an image processing module or engine for enhancing the visual quality of images. Techniques implemented by software 14 include multiple methods for video noise reduction, masking, smoothing, sharpening, contrast and histogram adjustment, contrast stretch and contraction. Software 14 also analyzes the image and adjusts the algorithms to optimize the image based on predictable procedure, specialty, surgeon, or other predictive factors. Software 14 also uses multiple analyses of the source image to dynamically adjust the image processing parameters.

Files created by system 10 are stamped with an internal "authentication" code not accessible to the user to designate the file as uncorrupted or structurally changed for legal purposes. Any image process that would substantially alter the information contained in the file causes this stamp to be modified, thereby indicating that the file is no longer primary source documentation. The utilities provided to cause image modification provide an opportunity to copy the image prior to modification so

that alterations can be performed without compromising the evidentiary quality of the documentation.

Referring to Fig. 11, as discussed above, video subsystem 34j processes images obtained during endoscopy for display on virtually any type of display device and in any suitable format. Preference database 20 supplies the conversion information that specifies the file type of the output image and the process to be performed (step 170). Video subsystem 34j uses this information to process the raw image data 172 (step 174). It is at this point that the authentication code is applied to the image file (step 176). If desired, external conversion utilities are applied to the image file (step 177). The image output file 178 is then processed for display (step 179) in any of the formats listed in Fig. 11. Processing occurs via a process of color reduction, whereby in the transition from a high (e.g., 24) bit depth to a lower bit depth (e.g., 16, 8, or gray scale), the process is one of calculated selection of specific color representations to be retained (as opposed to arbitrary file truncation).

Referring again to Fig. 2, controls of a VCR 66 are programmable using one or more data input devices 22 (e.g., remote control 48). Signals from remote control 48 are interpreted by a command interpreter in remote control adapter 34c according to preferences stored in databases 20 which associate actions taken by remote control 48, such as the actuation of one or more buttons on the device, with a command). The command interpreter relays the appropriate command (e.g., for record, stop, pause and play) to VCR subsystem 34m, which in turn transmits the commands to VCR 66, either via a hardwired connection or via interpreted and generated infra red signals.

A provision is also made for support of a system where frame by frame address information is written onto the video tape. Frame addresses are stored in the imaging system database, and can be used to index, record and playback videotape on a frame accurate basis.

System 10 is also capable of interactive recording or playback of audio signals. The audio material is written to and played back from digital files by an audio adapter subsystem 34 (not separately shown in Fig. 2).

Referring also to Fig. 12, the operation of system 10 in response to the remote control devices that can be used with the system is shown. As discussed, the remote control devices include infrared remote control 48 and voice pickup 50; other remote control techniques, such as detection of pupilary eye movement or the closure of contacts on, e.g., a footswitch, may also be used.

In response to an action taken by the operator, such as the depression of a button on IR remote control 48 (step 180), the subsystem 34 associated with the device (in this example, remote control adapter 34c) detects the signal (step 182) and interprets the signal (step 184) using information stored in an internal memory (not shown). Subsystem 34c then accesses preference database 20 (step 186) to determine the action to take in response to the operator's command. The action is communicated to the device driver for the device that is impacted by the command (step 188), and the action is performed. Thus, a generic action taken by the operator is translated to a device-specific instruction according to the preferences of the particular operator stored in preference database 20.

Referring to Fig. 13, system 10 also performs a slide-making function 190. Slide function 190 allows:

1. rapid and easy integration of high quality digitally processed images acquired using the OR image capture system discussed above; and

2. a method for constructing presentations. A presentation consists of a series of slides that contain formatted layouts of image, text and graphic information.

Image information may be used by many presentations, and stored only once. The system database (discussed below) allows for this by supporting "one-to-many" and "many-to- many" data relationships. Slides, images on slides, or any other element comprising a slide may be shared by greater than one presentation. An object will continue to exist until all presentations that call it are modified or deleted.

Slide function 190 performs the following operations:

Image processing functions

This part of the system will allow more advanced access to manual control of digital image processing functions described.

Scaling and positioning

Multiple images may be scaled and positioned on a single slide. The appropriate scaling and positioning will be done on the fly from the appropriate source image, leaving the source image unchanged.

Text

High quality antialiased text will be available in multiple fonts and point sized for overlay and text functions.

Preview Mode

Image information will be displayed in a "rough cut" version for processing speed. A presentation may be viewed as multiple miniature representations of the component slides on a single screen. The slides can be moved or modified using any pointing device to

restructure the presentation. This can be done on a laptop, personal computer, pen based system, or workstation.

Batch Print

A presentation may be queued for printing. The system will automatically control a digital film recording device, transferring the digital files to film. Automatic recording of number of available frame and errors will be incorporated. This should allow for most slides to be printed and discarded, allowing all filing to be conveniently located on the machine.

Preview and Edit

A simple interface allows the operator to view and edit slides in a so-called "what you see is what you get" presentation.

Referring to Fig. 13, the device handling architecture 192 implemented by system 10 is illustrated. This architecture allows many devices 16, 18, 22, 26 (Fig. 1) to be supported by a single operating program 14. Custom configurations are stored in a taxonomy in preference database 20, which is referenced during the configuration and operation of system 10. This architecture, and the structure and organization of the database (described below) allows system 10 to be platform and device independent -- that is, when a device is added to or removed from the system, only the software driver that controls that device need be changed. Portions of the programs or device drivers may reside in host processor 30, the device itself, or on other media or memory accessible to host processor 30 and the device.

Fig. 15 illustrates the connectivity and operation of a PCMCIA adapter 196 (e.g., adapters 34f-34i and 34p of Fig. 2). PCMCIA is an evolving standard for an array of devices that include storage, communicating and other external functions. In system 10, the PCMCIA (or any

other suitable standard, such as JEIDA) is used to interface to multiple input and output devices in a standard manner. PCMCIA adapter 196 is self-contained and durable, and is equipped with a standard edge connecter for ease of use. No additional wiring is required; the adapter slot provides operating power. Thus, the action of inserting adapter card 196 into the slot in host processor 30 is all that is required to initiate appropriate identification and management of the device that is inserted.

A PCMCIA adapter 196 may alternatively be located externally to host processor 30. For example, a PCMCIA adapter 196 may be disposed in another device (such as a laptop or palmtop computer 54, Fig. 2, that communicates with host processor 30.

Fig. 16 illustrates options 200 provided to the user by preference database 20 and shows how the user and CPU 38 interact with preference database 20. Options 200 include specifications as to video format 202, processing 204, the video sources to be used 206, and the configuration of the hardware 208. In addition, information concerning the users 210 and the hospital (or other facility in which system 10 resides) 212 is also provided. The attributes of the stored data 214, relationships between the data 216 (e.g., one-to-one, one-to-many, many-to-one, and many-to-many) are also provided, as are calculated data field attributes 218 (such as required entry, text, date, numeric, or decimal; other types of data fields are date and time).

Different physicians who use system 10 typically have different preferences as to how endoscopy images are to be obtained, processed, stored, and displayed. Indeed, a given physician's preferences for these options may vary according to the procedure being performed, or even according to the particular stage of any one

procedure. Preference database 20 classifies the preference information according to the individual physician, specialty, procedure, and stage of procedure to give the physician the capability of configuring and operating the devices in system 10 rapidly, according to his or her pre-stored preferences.

When the physician begins the procedure, the physician enters his or her name, specialty, and any other information that corresponds to the physician's desired setup, e.g., identification of the procedure to be performed (step 220). CPU 38 retrieves the preference information for this physician and procedure and uses it to configure the devices (e.g., cameras, CRT displays) preferentially used by the physician during the procedure (step 222). CPU 38 also maintains and updates the preference information in response to changes ordered by the physicians.

Figs. 17 and 18 illustrate procedures for storing images in and retrieving images from an image database. The image database can reside in any suitable memory device in system 10 (e.g., disk storage).

<u>System Software -- Overview</u>

The external view of system 10, to the end user, is that of a medical specific application, designed to handle both the capture and storage of real-time data, as well as the maintenance and manipulation of the data and relationships of the data to other associated information. This external view can be represented by Fig. 19.

As shown by Fig. 19, the end user first encounters a login procedure 240, which admits him or her to system 10. The user may then go to either Data Acquisition (Procedure) 250 or Information Processing (Query) 260. Both of these two sub-systems can invoke a signal processing subsystem 270 and an output subsystem 280.

The primary non-medical features of program 14 (Fig. 1), from the end users view, include:

1) Simplicity of use

Program 14 has been designed to be as simple as possible to the end user. This means that program 14 uses heuristic approaches, in many cases, to avoid having to ask complex questions to the user.

2) Platform Independence

Program 14 is designed to run on a multitude of platforms, and to allow many forms of input and output. This allows program 14 to be easily maintained, upgraded and expanded, and is not limited to availability of specialized equipment.

<u>System Software -- Technical Description</u>

Program 14, as illustrated in Fig. 19, embodies a much larger set of engines (i.e., software modules) running under the surface. Each engine is designed to provide a fairly easy mechanism for maintaining a flexible environment. Flexibility includes the ability to make major changes in both the functionality and presentation of the software, as well as to provide platform independence for easy integration of new software and hardware.

Referring also to Fig. 20, program 14 is divided into several engines each of which is devoted to a particular portion of the application, and is designed to integrate well with the other engines in the application. The engines included in this application are described below:

MOM 300	Menu management engine
EMM 310	Memory management engine
RA 320	Raster engine
DAD 330	Platform independent display engine
THE 340	Input device engine
GENERAL 350	general handlers for system independence

IGGY 360 Signal processing engine

POP 370 Output device engine

SIS 380 Database engine

Fig. 20 shows the interconnectivity of the engines, and their overall location within the application.

The following sections describe in detail each of the engines, its purpose and features:

Menu Manager Engine (MOM) 300

Menu management engine 300 handles the creation and maintenance of the entire software application that executes in system 10. To control MOM 300, the application describes menus as a series of encapsulated data arrays. The data arrays embody the following information:

1) Top Level Menu Array
    - A) Menu Location
    - B) Menu Size
    - C) Menu Background Graphics
        - a) RA Graphic token list
    - D) Menu Help Data
    - E) Menu EMM memory address
    - F) Menu efficiency tables
    - G) Menu Button Information
        - a) Button Count
        - b) Button ID
        - c) Button Location
        - d) Button Size
        - e) Button Graphics
            - 1) RA Graphic Token List
        - f) Button Function Pointer
        - g) Button Attributes
        - h) Button help text

With this technique, the entire control path of the application is described in data arrays, instead of

embodied in code. This flexibility allows the program control, as well as the look and feel of the program, to be changed by simple manipulation of data, instead of with code. Furthermore, this technique allows the program itself to make critical control path and look/feel changes, on the fly, while the user is running it. This capability allows the users to custom design the program to their tastes, affording an individualized fit to each and every user. The only thing the applications programmer needs to provide is the controlling data arrays (described above) and the functionality (action) for each "button" described in those arrays.

MOM 300 is designed to handle any number of "windows" on a display screen. The above data arrays describe both the look and feel, as well as the functionality of each window. A "window" may consist of a background and one or more buttons. A background describes how the window appears, and is not limited to any particular look. A button is an entity which exists on top of the background, and provides either information or functionality. When providing information, the button retrieves some data and displays it to the user in an understandable manner. When providing functionality, the button causes some action or actions to occur when the user "pushes" the button (e.g., by touching the button on a touchscreen or activating it with a mouse).

MOM 300 uses an integrated help system to allow the user to query both windows as well as buttons to see thier functionality or use before being required to actually use the system. As can be seen by the above array, the help system is embodied in the menu and button descriptions themselves, and thus afford easy change and upgrade with the software.

MOM 300 allows the display of windows to overlap, and handles complete maintenance as to the order and depth of windows, as well as the displaying of the windows which become revealed through use.

In addition to the multi-window approach used by the menu manager, MOM 300 also provides multi-processing on the window level, as the user is never restricted to working only on the "top" window. Through dynamic array switching, MOM 300 allows the user to select any visible button, regardless of the associated window or its depth in the screen. Because of this, MOM 300 implements a flat window scheme where no window has precedence over any other window.

By providing this level of management, the ease of use to the user is greatly improved by always allowing the user to "have their way". However, to provide total flexibility, MOM 300 does include an ability to simulate the more traditional hierarchical menu structures, if the need arises.

MOM 300 performs its work in a fairly straightforward manner. When an application beings to run the first thing done by the program is to invoke MOM 300 routines to open one or more windows. Once this is accomplished, the program then invokes MOM 300, which takes complete control of the program from that point on.

MOM 300 analyzes the existing menu structures and begins to query the users requests via THE engine 340. Display of the menus, as well as display of the users input device and selection of choices, is handled within MOM 300 by calling RA engine 320 to draw the desired information, and DAD engine 330 to display it to the current output device.

No rules are placed on what can be performed by the functionality invoked by MOM 300 via the data structures. If desired, windows which were opened at the

beginning of the program may be closed, and new windows opened. Likewise, any number of additional windows may be opened. These actions may occur in response to users activating various functions on the existing windows. This may even include the restructuring of buttons and data within existing windows. For example, the user may select a button which causes half of the buttons on their current window to be replaced with an image. This type of dynamic remodeling of the menu manager environment is totally handled by MOM 300. The functionality merely expresses the changed data arrays and MOM 300 handles the reintegration of the data into the existing environment in a seamless way.

MOM 300 is also responsible for all event handling at the menu level. MOM 300 can be made to poll events at every input device cycle, and maintains, among other things, a real time clock for timing of events and maintenance of a time-of-day clock. MOM 300 uses THE engine 340 for event polling, but allows button functionality to uncouple event polling for specialized event handling within a buttons action.

MOM 300 never terminates, unless all windows are closed without a new window being opened prior to functionality return. Besides this method, the only other method of termination is direct exit within a function.

Memory Management Engine (EMM) 310

EMM engine 310 is responsible for the allocation, maintenance and freeing of the data arrays used by RA engine 320 and the other functionality portions of the program. Despite the fact that modern programming software provides memory allocation, the use of EMM engine 310 assures platform independence when moving between software platforms as well as hardware memory platforms.

EMM engine 310 is a fairly simple code system which allocates memory in "pages" or "blocks" when requested by the rest of the software. EMM 310 maintains data lists to track the memory, and provides boundary checking as well as freeing of the data when its use is complete.

Finally, EMM 310 also provides the ability to query the memory system for remaining quantity and availability, allow the software to provide status, as well as predict remaining capability.

EMM 310 is used by MOM engine 300 to allocate the memory used for the windows (see the section below on RA engine 320 for a more detailed description) as well as being available to all the other engines as needed.

Raster Engine (RA) 320

RA engine 320 is the main drawing engine responsible for drawing the menus and their associated buttons and backgrounds. Because this software system is designed to be platform independent, RA 320 does not draw to any output device. Instead, RA 320 interfaces with EMM engine 310 and draws to EMM allocated memory. Actual display of the windows on the users display device is handled by DAD engine 330. This type of architecture is termed a virtual frame buffer.

RA 320 is driven directly by MOM 300, and in some cases by the application functionality itself. RA 320 is a table dispatched service and takes as input a pointer to a heterogeneous data array. This data array consists of pointers and data, where the pointers correspond to a token list which embody a primitive graphics language.

The token list described by RA 320 contains the minimum needed set of drawing devices to render all menu drawing. This includes gradient rectangles, antialiased and non-antialiased text, circles and lines, as well as more complex concepts such as buttons, sliders, controls,

LED's, and textured backgrounds. In order to provide capabilities which may be needed later in the development cycle, RA 320 also provides programmer definable tokens, via a functional token, with which the programmer can vector to thier own code during a RA token execution. This enables the program to draw any type of object, even objects never conceived before in the software.

RA 320 is limited to drawing to a single window at a time, and many not mix drawing to different windows within a single token download. Therefore it is up to MOM 300 to direct RA 320 as to the appropriate window for drawing, as well as providing RA 320 with the correct token list, as extracted from the menu arrays.

Because of the self-modifying nature of the software, as well as its platform independence, RA 320 is designed to allow self-modification during the execution of a RA token list. In other words, a token list can self-modify while executing, allowing decisions to be made on the fly, as to future drawing which is still down-stream from the current point of token execution.

RA 320 always draws via EMM engine 310, in a standard method. While the method is modifiable, it is a fixed preapplication, as per the programmer's judgment. For example, within this application, RA 320 is defined to be 24 bit (8 bit per pixel) in a Packed architecture (RGBRGBRGB ...). By splitting RA 320 and DAD 330 apart, and allowing DAD 330 to handle device particulars, DAD 330 can color reduce/expand, as well as resolution reduce/expand the RA maintained EMM memory buffers when displaying to any device.

Font drawing is handled in RA 320 by use of pre-computed raster font tables. The font tables contain a byte per pixel, which indicates the natural antialiasing of the font at that pixel. Fonts are encoded such as to be both proportional as well as non-proportional, and any

font may be displayed in any combination of antialiasing and proportionality. When fonts are displayed, an additional font transparency sets the level of opacity, allowing for drop shadows and other effects. Font color is specified as a 24 bit (RGB) value, which is used with the transparency and antialiasing information to calculate the appropriate color for each pixel. The software allows any number of fonts to be loaded concurrently, and fonts may be switched dynamically, on a character by character basis if desired. Font concurrence is only limited by available memory (as all fonts used are always resident).

Platform Independent Display Engine (DAD) 330

DAD engine 330 handles actual updates of information to the user's end display device. Since the software system is device independent, DAD 330 coordinates requests from MOM 300 and the user's functionality, and reads EMM memory and displays the memory to the current device.

Basically the cycle works as follows. MOM 300, following advice from the menu structures, instructs RA 320 to draw the various windows to EMM memory. After RA 320 completes the drawing, MOM 300 then instructs DAD 330 to take the completed memory and display it on the current display device.

When DAD 330 receives an instruction to display a piece of EMM memory, it consults the MOM structures to determine the location of the memory, and the desired location of its display. Because DAD 330 is purely BLIT oriented (e.g., high speed rectangles), the drawing is both efficient and space saving. Since all display devices have inherent differences, it is DAD's responsibility to call the appropriate routines at the proper time, for each display device.

When a new device is added to the system, only a small handful of routines need to be supported for the device. Of the total roster of routines, only routines which the display device is capable if need be supported. Routines which are not possible for that display device need not be stubbed, and DAD 330 is simply informed of thier absence and handles it gracefully. The routines required embody a small set of primitives, which include area blitzing, line blitzing, high speed rectangle drawing, high speed or outlined rectangle drawing and cursor support. Other routines, not necessarily supported by all devices include, acquisition, high-speed unpacked blitzing and automatic packing and unpacking routines.

Because the amount of code for support of a driver is relatively small, the drivers are compiled directly into the software. This capability allows DAD 330 to actually maintain and output to multiple devices simultaneously, making it attractive for multiple displays as well as switching between hardware on the fly.

In order to make DAD 330 handle the platform independent drivers, the software calls a standardized set of routines, which are table dispatched at runtime. Because of the table dispatch method, no comparisons are required to locate a routine within a driver. Instead, all execution is handled by hashed indexing which provides near-instantaneous access to all drivers an all functions. Note that because DAD is responsible for both input and output display devices, it handles any video acquisition required by video data software within the user's procedure program. This requires that DAD 330 coordinate with SIS 380 and MOM 300 at the proper time, for control and decoding of incoming video.

Input Device Engine (THE) 340

THE engine 340 is designed to be a device independent input event handler. THE 340 is invoked by both MOM 300 and the users functionality, to retrieve spatial and textual information from the user. Because the software is input device independent, THE 340 is responsible for the combination of multiple pointing devices, and can handle more than one input device simultaneously, as well as being able to switch input device context on the fly.

THE engine 340 is basically divided into two sections: textual and spatial. The textual handler involves reading textual data from keyboard devices and transmitting them, when requested, to MOM 300 and functional software. Textual handling includes not only input and maintenance of text buffers, but also flushing of all textual queues.

Spatial handling is performed by determining the location and types of input devices. Because input devices fall into two categories, absolute and relative, THE engine 340 automatically converts all coordinate, regardless of their type, into absolute coordinates. These coordinates are automatically clipped, by THE engine 340, against the current screen resolution as defined by DAD engine 330.

One aspect of THE engine 340 is its coupling with MOM 300 for mouseless handling of menu structures. Because MOM 300 allows menus and buttons to exist anywhere on the screen, even layered, the software which allows the user to move from "field" to "field" (including buttons) must be able to dynamically analyze the "best next" button to go to, from the current button. This is especially important when considering the fact that the user may modify the look for the o menus, on the fly. Therefore, a fixed rule for moving fro button to

button via cursor keys is out of the question. THE engine 340 handles this, with help from MOM 300, by keeping track to two cursor locations. The current absolute cursor, and the current field cursor. The field cursor marks the current (or most recent) choice executed by the user (e.g., the last button picked). When a cursor movement command is detected by THE 340, it uses MOM engine 300 to retrieve the menu descriptions, and analyzes these descriptions in an attempt to determine the best location to move to, and then performs the move to that location. This form of dynamic sensing affords the most flexibility to the entire application.

General Handlers for System Independence (GENERAL) 350

GENERAL engine 350 handles all routines which are fairly standard, but which for one reason or another are desirable to have under platform independent control of the application. This includes items such as non-EMM engine memory acquisition and freeing, as well as file I/O. GENERAL 350 is engine present to give a simple and global mechanism for error control and handling. This provides the software package with a uniform mechanism for error recovery, which otherwise may be handled in a less than elegant way by the operating system of run-time program.

Signal Processing Engine (IGGY) 360

IGGY engine 360 is responsible for all digital signal processing of acquired wave form (i.e., image) data. IGGY 360 is commanded primarily by MOM 300 and the functionality used to perform some form of wave form processing (usually image processing). Instead of being an integrated engine (as are MOM 300, RA 320, DAD 330, and the other engines), IGGY 360 is a set of separate routines, which may be used and combined to perform high level processing of acquired data.

Due to the nature of the application, in that the end user does not possess the education necessary to perform high-end signal processing, IGGY 360 is designed to have heuristic processes which allow it to analyze wave from data and determine the best enhancements necessary. IGGY 360 can use MOM 300 to request minimal information from the user, but in general does not do so except in the simplest of cases.

IGGY 360 embodies at least the following basic signal processing techniques:

1. Histogram Analysis
2. Histogram Stretching
3. Histogram Equalization
4. Noise Analysis
5. Noise Reduction
6. Area of Interest Location
7. Unwanted Area Removal -- i.e., rather than compress the data, only meaningful data (such as the picture of the surgical site) is retained, while other data (such as borders that contain no real picture information) are discarded
8. Sharpening
9. Blurring
10. Feature Extraction
11. Feature Highlighting
12. Blending
13. Artifact Removal
14. Non-linear balancing
15. Colorization.

Though IGGY 360 provides these services in an on-call manner, the analysis services utilize heuristic techniques to determine if the other services are required to provide the best output data possible.

Output Device Engine (POP) 370

POP engine 370 is to the output devices as DAD 330 is to the input devices. For each output device

supported, POP 370 provides interface mechanism for that device. POP 370 also provides limited input from some of these devices, as the device requires. POP 370 is responsible for maintaining output (and sometimes input) to the following devices:

1. Printers (color and black and white)
2. Film Recorders
3. Data Recorders
4. Tape Drives
5. Disk Drives (hard and floppy)
6. FLASH cards (PCMCIA)
7. Networks
8. Modems

It can be seen from the above list that POP 370 is responsible for some diverse device handling. POP 370 places the devices into "like" categories, and provides a uniform mechanism for calling "like" devices. For example, all drives, flash cards and networks are driven in a like manner, while modems have their own mechanism, just as printers and film recorders. Since MOM 300 directs the output activities, POP 370 is never confused as to how the service is intended (e.g., each service is specialized, and thus POP knows the appropriate need and duty).

While most of handling provided by POP 370 requires simple interfacing to outside drivers, or the operating system, the printer and film recorder interfaces of POP 370 are highly particular to the device being addressed. Furthermore, because of all the possible formats for output devices, POP 370 interfaces with MOM 300 to provide the user with a visual mechanism for the specification of output format. For example, under printer interface, POP 370 can direct MOM 300 to display a grid of images, and have the user select the appropriate image and its location. Furthermore, POP 370

interfaces with RA 320 to allow the user to annotate the images prior to output. POP 370 handles interpolation of incorrectly sized data, and can utilize in-memory fonts as well as printer provided fonts, depending on the hardware capabilities.

Database Engine (SIS) 380

SIS engine 380 is the database handler. It is responsible for handling all database activities which include user management, forms management, data management and signal processing management.

User management is invoked by MOM 300 to confirm a user and to set a users profile for the display of MOM menus. Forms management handles the design, storage and retrieval of forms. Data management handles the storage and query of data from within forms. Signal processing management is a subset of data management, and is responsible for the handling of signal data within a form.

Overview of Flow Control

As discussed above, program 14 is designed to be a specific medical application, even though the engines provide a much greater, and universal capability.

The overriding flow control is a section of the software which dictates to MOM 300, POP 370, SIS 380, and IGGY 360 a particular method to follow for user flow control. It is this overriding flow control which provides much of the automatic handling which is required by the naive user.

Referring also to Fig. 19, when the user invokes the program 14 at login time, the user is greeted by an opening screen and requested to enter a user name and password. Once entered, these elements are cross compared, via SIS 380, against the valid users in the database. When a valid user is identified, his user

profile is loaded, and MOM 300 is informed of any changes to the environment.

If the user has an associated procedure form with his or her login, the user is asked whether this is a procedure. If the answer is affirmative, procedure flow control 260 is followed; otherwise, information flow control 250 is followed.

Procedure Flow Control 260

Once procedure flow control 260 is accessed, the procedure handler accesses the user's procedure form and displays it to the user. This is some form, entered for or by the user, when the user is given original access to the software. The form which is assigned to the user may be modified at any time via the information processor.

Once the user fills out the form, the procedure flow control 260 is launched. This causes DAD 330 to invoke acquisition and control is switched to the user. THE engine 340 monitors a special remote control, used during procedures (or, optionally, the keyboard), and causes various video functions to be executed. These include, but are not limited to:

1. Zoom IN and OUT of the live video image
2. Freezing the live video signal, and then discarding it
3. Freezing the live video signal, and then saving it sequentially via POP 370
4. Reviewing all currently acquired video signals
5. Executing a TOUR (directed capture sequence)
6. Adjustment of the input video parameters

Captured images are sequentially numbered and named, and are stored on the current mass storage device via POP 370. At the end of the procedure, SIS 380 is automatically called to store all, or only approved images into the database, associated with the current form.

During the REVIEW process (122, Fig. 6), the user may review all captured images, and optionally delete unwanted images or invoke special image processing on selected images. Images selected for special image processing are processed automatically and overwritten with the new results.

Upon exiting procedure 260, the user is given the ability to download the acquired data to a removable media device (e.g., FLASH memory, etc.), and then is logged off the program.

<u>Information Processing Control 250</u>

The information handler is invoked upon successful login, when the user either does not have an associated procedure, or does not wish to access procedure. Once in the information processing system, the user is first greeted with a list of available forms (via SIS 380). The user may select an existing form, or select a "new form" option to create a new form. If the user opts to create a new form, the various form elements which are available are shown to the user, plus the ability to create new form elements. A form element is any item when exists on any other form.

When the user picks an existing form, the form is loaded with blank data. The user may fill out one or more of the fields in the form, and either enter the form into the database (e.g., enter the instance of the new data) or else request a query. When a query is requested, SIS 380 will attempt to fulfill the missing elements of the form, based on the fields which have been filled in. Upon retrieving the list of available responses, the user is shown the list and allowed to pick an item from the list. Once picked, SIS 380 will load the form with the appropriate data, and will display it to the user. Images, or other signal processing data within the form, when selected, will invoke a image

browsing menu which displays all associated images with the form. Further selection of an image will cause IGGY engine 360 to be accessed for full signal processing by the user.

Information processing procedure 250 also allows the user to format and output query responses to the printer, in a variety of looks, all handled by POP 370.

Finally, for system administrative users, information processing procedure 250 allows new users to be added to the system, and existing users to be removed from the system.

Database Configuration

Referring to Fig. 21, preference database 20 and all image databases are organized in a manner that maximizes the flexibility of adding, modifying, or deleting information from the databases at will, without rebuilding or restricting availability of data. The structure of the databases is relatively simple, which keeps the data model independent of features of host data storage. In addition, database subsets (datasets) can be derived, extracted and manipulated remotely and then merged back.

Each data element is stored independently in database storage. That is, there is no record relative pointer structure in the dataset. All "pointers" are data relative. One advantage of this arrangement is that if a database file is damaged, pointer chain corruption (which could damage other stored data) is avoided.

A dataset may be transformed into plain text to allow for portability.

Record Structure

Dataset storage is implemented using two record types, the "what" record 400 and the "it" record 410. "What" records 400 describe the data dictionary, while "it" records 410 contain the actual data.

Consider the following sample schema definition (in this case, using the RAIMA database):

```
record  what {                                  /*field types*/
        unique key long what_id;  /*field key*/
        key    long what_owner;   /*key of its owner*/
        char   what_name[32];         /*field name*/
        long   what_look;         /*it's appearance*/
}
record  it {                       /*data item*/
        unique key long its_id;           /*its key id*/
        key long its_owner;        /*key id of its owner*/
        key long its_what;         /*what kind of data is
                                     it?*/
        key char its_data[64];            /*the data
                                            itself*/
}
```

Each "what" record 400 and "it" record 410 has a randomly generated numeric key that has no special significance other than that it is unique to that specific record in the database.

The data dictionary (i.e., the mechanism that describes the types of data contained in the dataset) is held in "what" records 400. For example, a simple address book data set could require just two "what" records 400 -- one for a name and one for an address. The actual address book data is held in "it" records 410. One "it" record 410 is used for each name, and one "it" record 410 stores each address.

If the dataset needs to be expanded to include, for example, the telephone number of each address entry, a new "what" record 400 is created for telephone number. Then, any new entries in the database can also include an associated telephone number -- but note that the existing data is not impacted at all. Consider the following example layout of such an address book database:

The "what" records 400 are:

A. 1024:0:name:1

(1024 is the random key; 0 is owner (none); "name" is the text name of the field; 1 denotes alpha string).

B. 3046:1024:address:1
(3046 is the key; 1024 denotes that address is associated with "name"; "address" is the text name of the field; 1 again denotes an alpha string).

C. 4096:1024:telephone number:2
(4096 is the key; 1024 indicates that telephone is also associated with "name"; "telephone number" is the text name of the field; 2 denotes this is telephone number field).

The address book entry "it" records 410 are:
1234:0:1024:John Doe
5748:1234:3046:911 Main St.
8674:1234:4096:(312) 555-1212

In the first it record 410, 1234 is the random key of the record, 0 shows this record has no associated "owner", 1024 indicates that this is a "name" record, and "John Doe" is the name data.

In the second it record 410, 5748 is the random key, 1234 associates this record with the first (John Doe) it record 410, 3046 indicates that this is an address record, and "911 Main St." is the address data.

Finally, in the third it record 410, 8674 is the random key, 1234 associates this record with "John Doe", 4096 denotes that this is a telephone number, and "(312)555-1212" is the phone number data.

The what/it structure allows several query methods. To find every data item associated with "John Doe" we search the "it" records data fields for "John Doe", read it's key field (1234), and then find all of the records that have 1234 in the owner field.

To obtain a list of phone numbers, we simply scan the "it" records looking for "4096" in the "its_what" field. If we need the name associated with these phone numbers, we look at the "its_owner" field (in John Doe's case is would be 1234) then locate records with 1234 in the "its_id" field -and- 1024 in the "its_what" field.

The what/it structure allows several types of data relationships to exist including one-to-one, one-to-many, many-to-one and many-to-many associations. All of these can be created dynamically simply by defining the appropriate "what" records.

Also note that no storage is taken for non-existent data. In the address book example, if there is no telephone number for a given entry, then no "it record is allocated.

That what/it method also lets us extract a subset of records by first obtaining all of the "what" records and then obtaining all of the desired "it" records by the "its_what" fields.

Other embodiments of the invention are within the scope of the following claims. For example, while system 10 has been described with reference to endoscopy, the system may also be used for other medical procedures, surgical or otherwise.

What is claimed is:

CLAIMS:

1. A system for acquiring images during a medical procedure and using the acquired images, comprising

at least one input device for obtaining said images,

at least one output device for using the images obtained by said input device,

a storage device for storing, for each one of a plurality of users of said system, information that indicates one or more processing operations to be performed on said images obtained by said input device, and

a processor, responsive to an identity of one of said users who is currently using said system, for performing processing operations on said images obtained by said input device and applying said images to said at least one output device based on the information in said storage device that corresponds to said current user.

2. The system of claim 1 wherein said information further indicates, for each one of said users, a configuration of said at least one input device, and further comprising means for establishing said configuration in response to said identity of said current user.

3. The system of claim 2 wherein said configuration includes a format in which said at least one input device produces said image.

4. The system of claim 1 wherein said information further indicates, for each one of said users, a configuration of said at least one output device, and further comprising means for establishing said configuration in response to said identity of said current user.

5. The system of claim 4 wherein said configuration includes a format in which said at least one output device uses said image.

6. The system of claim 1 wherein said information further indicates, for each one of said users, a sequence of images that are to be obtained during said procedure, and further comprising means for prompting said current user to obtain the images in said sequence.

7. The system of claim 1 wherein said storage device further stores said information for a plurality of surgical procedures, said processor being further responsive to an identification of a current one of said surgical procedures and performing said processing based on said information that corresponds to said current surgical procedure.

8. The system of claim 1 further comprising a plurality of said input devices, said storage device storing said information for each one of said input devices, said processor selectively processing images from said input devices based on the identity of said current user and said information.

9. The system of claim 1 further comprising a plurality of said output devices, said storage device storing said information for each one of said output devices, said processor selectively applying said processed images to said output devices based on the identity of said current user and said information.

10. The system of claim 1 wherein said storage device stores said information as a plurality of linked records, a first one of said records identifying a type

of said information, and a second one of said records containing data associated with said type of information.

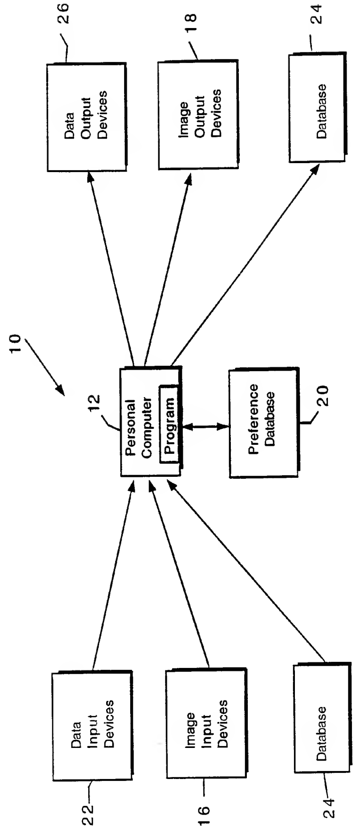

26
Data Output Devices
18
Image Output Devices
24
Database
10
12
Personal Computer
Program
Preference Database
20
Data Input Devices
22
Image Input Devices
16
Database
24

FIG. 1

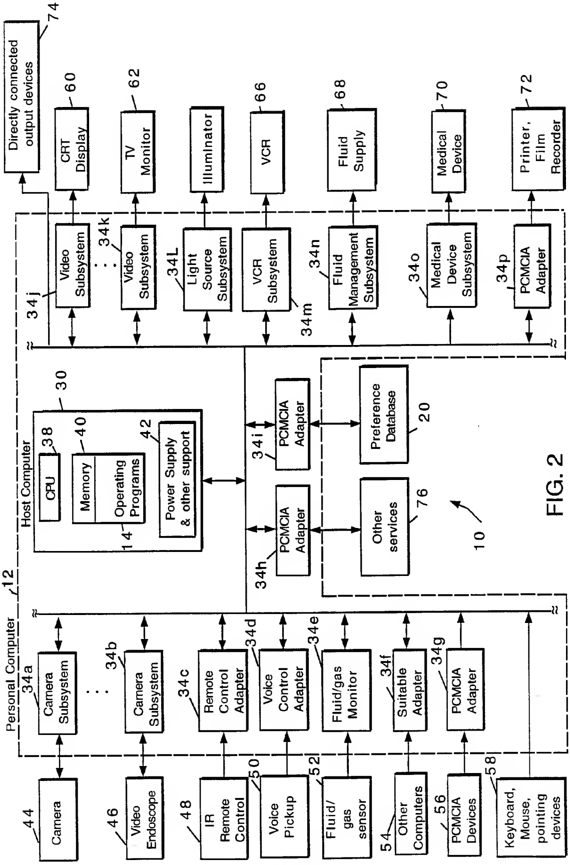

Directly connected output devices
74
CRT Display
60
TV Monitor
62
Illuminator
VCR
66
Fluid Supply
68
Medical Device
70
Printer, Film Recorder
72
Video Subsystem
34j
34k
Video Subsystem
34L
Light Source Subsystem
VCR Subsystem
34m
34n
Fluid Management Subsystem
34o
Medical Device Subsystem
34p
PCMCIA Adapter
Host Computer
30
CPU
38
Memory
40
Operating Programs
14
Power Supply & other support
42
34i
PCMCIA Adapter
34h
PCMCIA Adapter
Preference Database
20
Other services
76
10
12
Personal Computer
34a
Camera Subsystem
34b
Camera Subsystem
34c
Remote Control Adapter
34d
Voice Control Adapter
34e
Fluid/gas Monitor
34f
Suitable Adapter
34g
PCMCIA Adapter
44
Camera
46
Video Endoscope
48
IR Remote Control
50
Voice Pickup
52
Fluid/ gas sensor
54
Other Computers
56
PCMCIA Devices
58
Keyboard, Mouse, pointing devices

FIG. 2

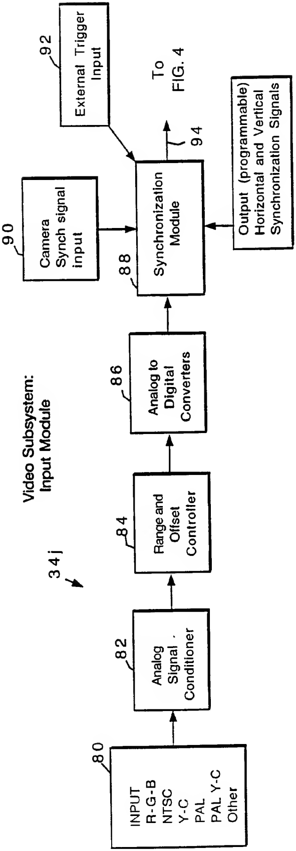

Video Subsystem:
Input Module
34j
INPUT
R-G-B
NTSC
Y-C
PAL
PAL Y-C
Other
80
Analog Signal Conditioner
82
Range and Offset Controller
84
Analog to Digital Converters
86
Synchronization Module
88
Camera Synch signal input
90
External Trigger Input
92
To FIG. 4
94
Output (programmable) Horizontal and Vertical Synchronization Signals

FIG. 3

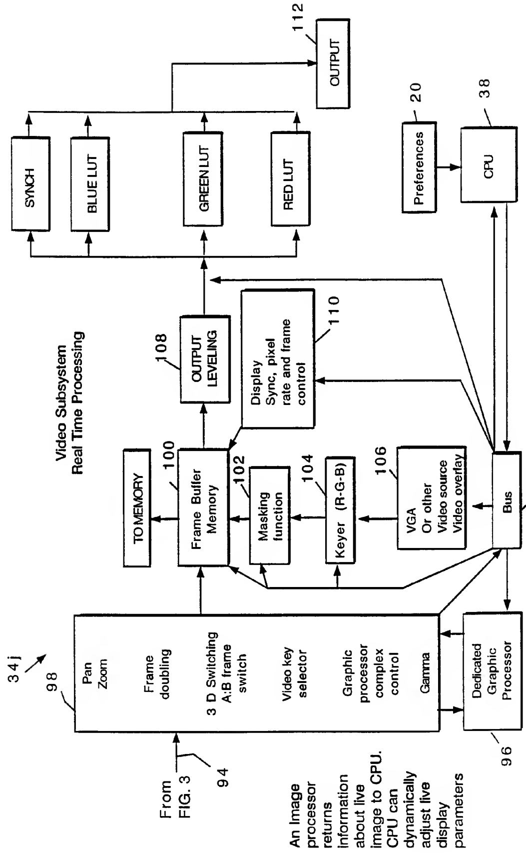

Video Subsystem
Real Time Processing
34j
98
Pan
Zoom
Frame
doubling
3 D Switching
A:B frame
switch
Video key
selector
Graphic
processor
complex
control
Gamma
From
FIG. 3
94
An Image
processor
returns
information
about live
image to CPU.
CPU can
dynamically
adjust live
display
parameters
Dedicated
Graphic
Processor
96
TO MEMORY
Frame Buffer
Memory
100
Masking
function
102
Keyer (R-G-B)
104
VGA
Or other
Video source
Video overlay
106
Bus
36
OUTPUT
LEVELING
108
Display
Sync, pixel
rate and frame
control
110
SYNCH
BLUE LUT
GREEN LUT
RED LUT
OUTPUT
112
Preferences
20
CPU
38

FIG. 4

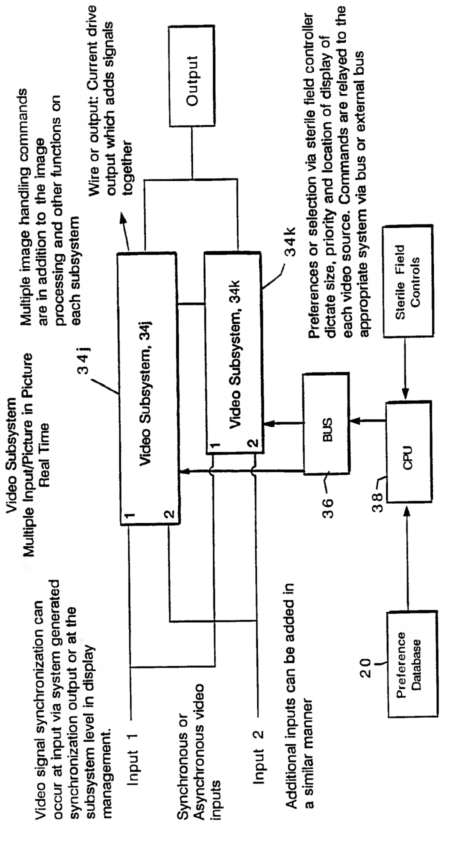

Video signal synchronization can occur at input via system generated synchronization output or at the subsystem level in display management.
Input 1
Synchronous or Asynchronous video inputs
Input 2
Additional inputs can be added in a similar manner
Video Subsystem
Multiple Input/Picture in Picture Real Time
Multiple image handling commands are in addition to the image processing and other functions on each subsystem
34j
Video Subsystem, 34j
1
2
Wire or output: Current drive output which adds signals together
Output
Video Subsysm, 34k
1
2
34k
BUS
36
CPU
38
Preference Database
20
Sterile Field Controls
Preferences or selection via sterile field controller dictate size, priority and location of display of each video source. Commands are relayed to the appropriate system via bus or external bus

FIG. 5

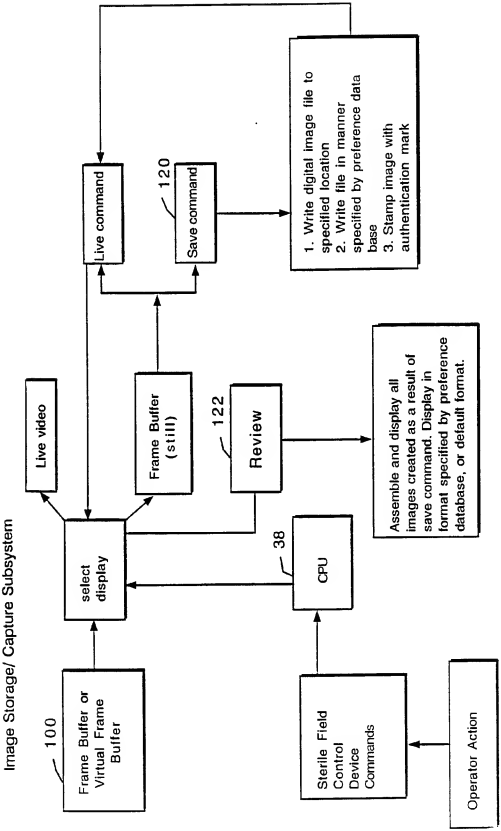
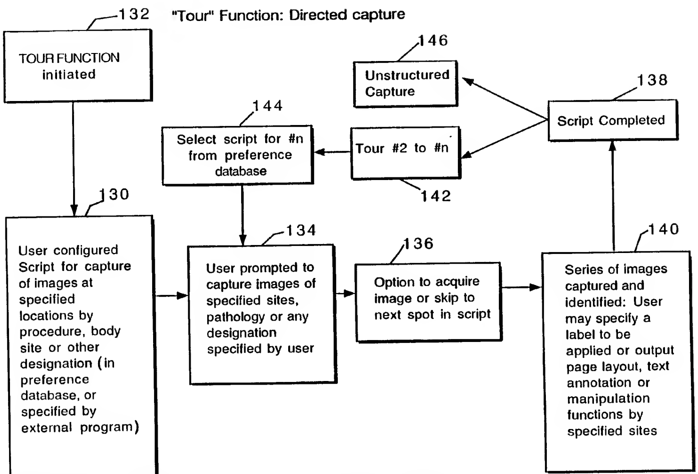


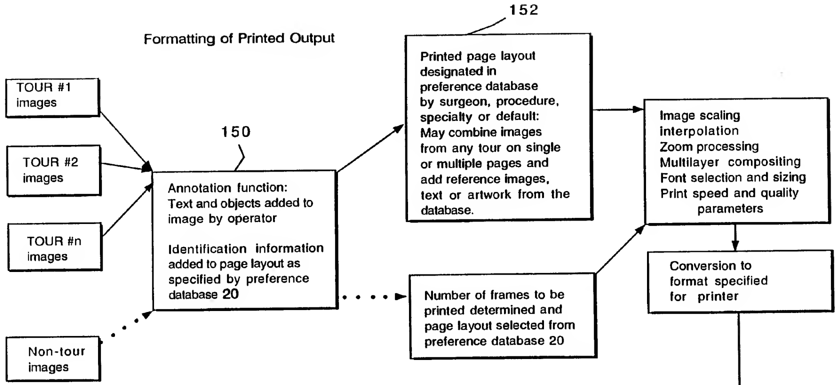
Image Storage/ Capture Subsystem
100
Frame Buffer or Virtual Frame Buffer
select display
Live video
Frame Buffer (still)
Live command
120
Save command
1. Write digital image file to specified location
2. Write file in manner specified by preference data base
3. Stamp image with authentication mark
122
Review
Assemble and display all images created as a result of save command. Display in format specified by preference database, or default format.
38
CPU
Sterile Field Control Device Commands
Operator Action

FIG. 6



c is provided such that a pre-op and post-op tour may be used to show
ology and post endoscopic procedure results or status. The output page
at may identify the sites as pre and post intervention.

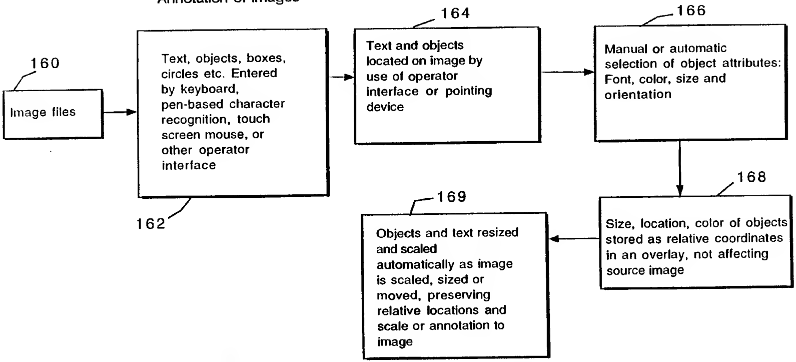
FIG. 7



Current Page description language is post script level 2. Other interpreters for conversion to other file formats can be added at any time.
Page formatting allows multiple images in multiple sizes to appear on a single or multiple pages.

FIG. 8

Annotation of Images



Annotation can be performed in a similar manner at both the image and page layout levels. A single image or page layout can possess multiple annotation overlays. Single or multiple overlays may be composited with the image. The overlay may also possess the attribute of translucency, ranging from obliteration of the underlying image, to transparency, in areas where the overlay is invisible. Single or multiple overlays may be attached to an image at a given time.

Printed output of the image/overlay composite will appear as if the overlay were part of the image itself.

FIG. 9

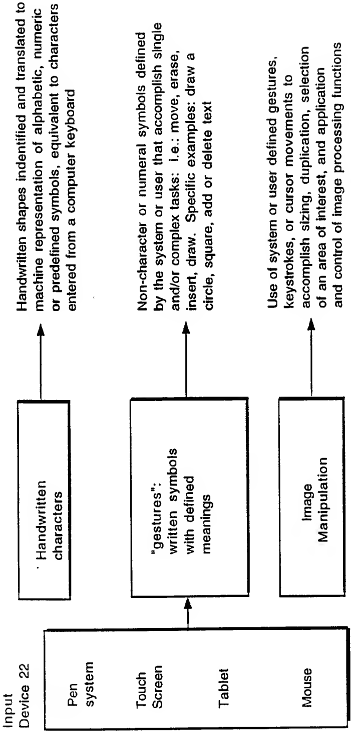

Input Device 22
Pen system
Touch Screen
Tablet
Mouse
Handwritten characters
"gestures": written symbols with defined meanings
Image Manipulation
Handwritten shapes indentified and translated to machine representation of alphabetic, numeric or predefined symbols, equivalent to characters entered from a computer keyboard
Non-character or numeral symbols defined by the system or user that accomplish single and/or complex tasks: i.e.: move, erase, insert, draw. Specific examples: draw a circle, square, add or delete text
Use of system or user defined gestures, keystrokes, or cursor movements to accomplish sizing, duplication, selection of an area of interest, and application and control of image processing functions

FIG.10

11/21

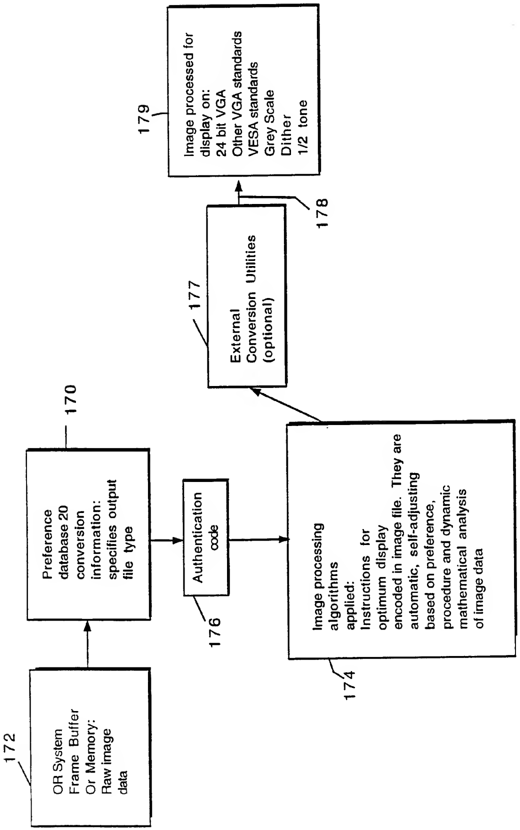

172
OR System Frame Buffer Or Memory: Raw image data
170
Preference database 20 conversion information: specifies output file type
176
Authentication code
174
Image processing algorithms applied: Instructions for optimum display encoded in image file. They are automatic, self-adjusting based on preference, procedure and dynamic mathematical analysis of image data
177
External Conversion Utilities (optional)
178
179
Image processed for display on: 24 bit VGA Other VGA standards VESA standards Grey Scale Dither 1/2 tone

FIG. 11

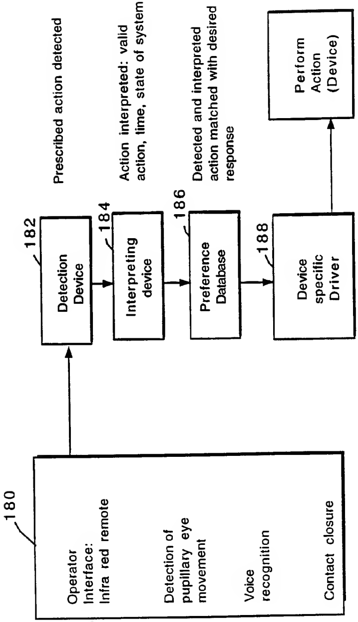

180
Operator Interface: Infra red remote
Detection of pupillary eye movement
Voice recognition
Contact closure
182
Detection Device
Prescribed action detected
184
Interpreting device
Action interpreted: valid action, time, state of system
186
Preference Database
Detected and interpreted action matched with desired response
188
Device specific Driver
Perform Action (Device)

FIG.12

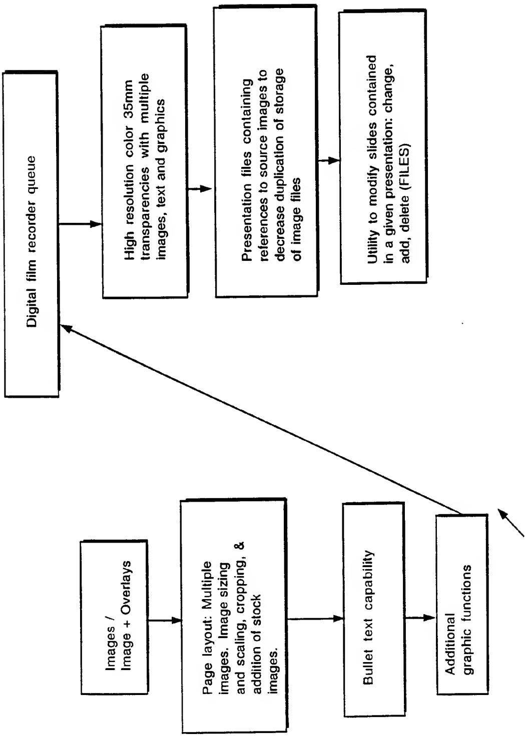

Digital film recorder queue
High resolution color 35mm transparencies with multiple images, text and graphics
Presentation files containing references to source images to decrease duplication of storage of image files
Utility to modify slides contained in a given presentation: change, add, delete (FILES)
Images / Image + Overlays
Page layout: Multiple images. Image sizing and scaling, cropping, & addition of stock images.
Bullet text capability
Additional graphic functions
190

FIG. 13

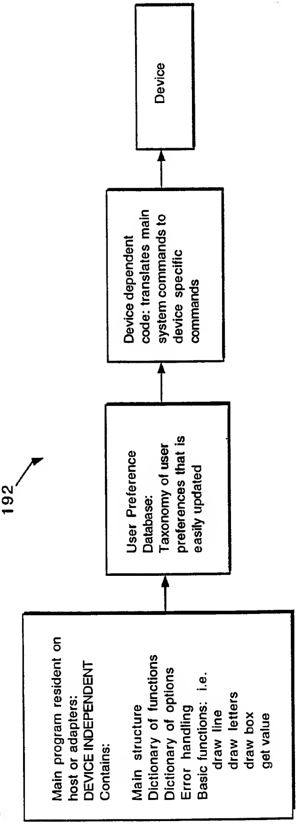

192
Main program resident on host or adapters: DEVICE INDEPENDENT
Contains:
Main structure
Dictionary of functions
Dictionary of options
Error handling
Basic functions: i.e.
draw line
draw letters
draw box
get value
User Preference Database: Taxonomy of user preferences that is easily updated
Device dependent code: translates main system commands to device specific commands
Device

FIG. 14

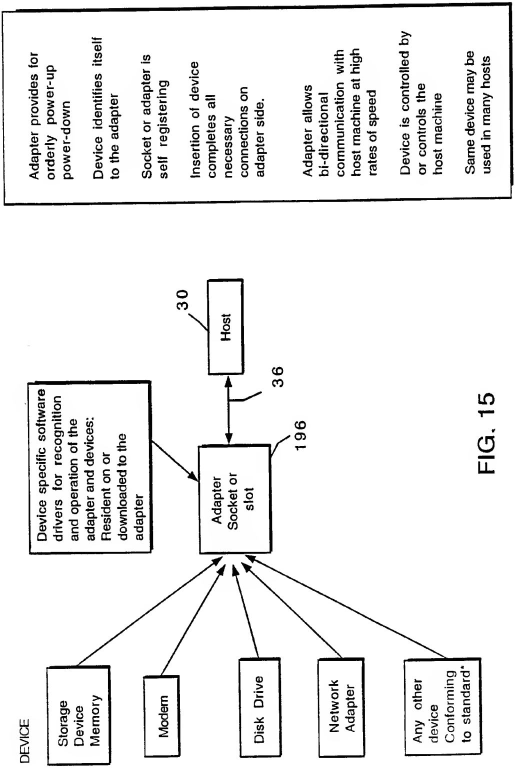

Adapter provides for orderly power-up power-down
Device identifies itself to the adapter
Socket or adapter is self registering
Insertion of device completes all necessary connections on adapter side.
Adapter allows bi-directional communication with host machine at high rates of speed
Device is controlled by or controls the host machine
Same device may be used in many hosts
30
Host
36
196
Device specific software drivers for recognition and operation of the adapter and devices: Resident on or downloaded to the adapter
Adapter Socket or slot
DEVICE
Storage Device Memory
Modem
Disk Drive
Network Adapter
Any other device Conforming to standard*

FIG. 15

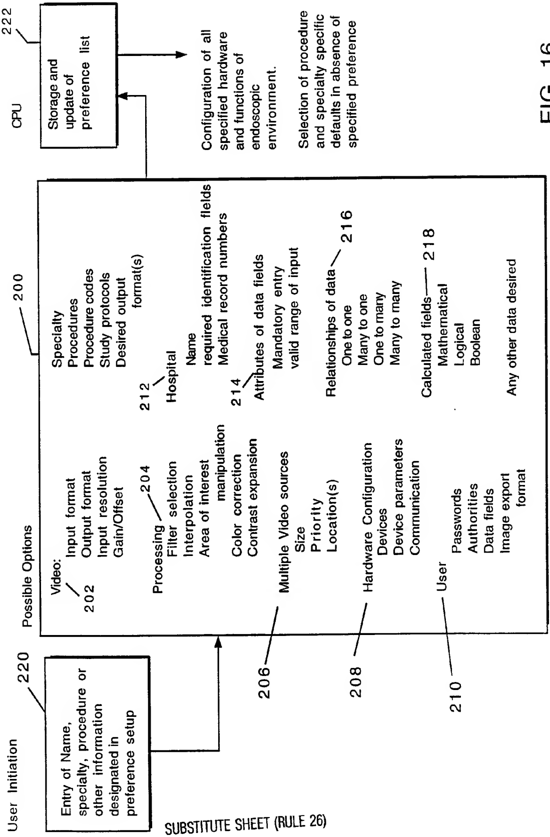

User Initiation
220
Entry of Name, specialty, procedure or other information designated in preference setup
Possible Options
200
Video:
202
Input format
Output format
Input resolution
Gain/Offset
Processing
204
Filter selection
Interpolation
Area of interest manipulation
Color correction
Contrast expansion
Multiple Video sources
Size
Priority
Location(s)
Hardware Configuration
Devices
Device parameters
Communication
User
Passwords
Authorities
Data fields
Image export format
Specialty
Procedures
Procedure codes
Study protocols
Desired output format(s)
212
Hospital
Name
required identification fields
Medical record numbers
214
Attributes of data fields
Mandatory entry
valid range of input
Relationships of data
216
One to one
Many to one
One to many
Many to many
Calculated fields
218
Mathematical
Logical
Boolean
Any other data desired
206
208
210
CPU
222
Storage and update of preference list
Configuration of all specified hardware and functions of endoscopic environment.
Selection of procedure and specialty specific defaults in absence of specified preference

FIG. 16

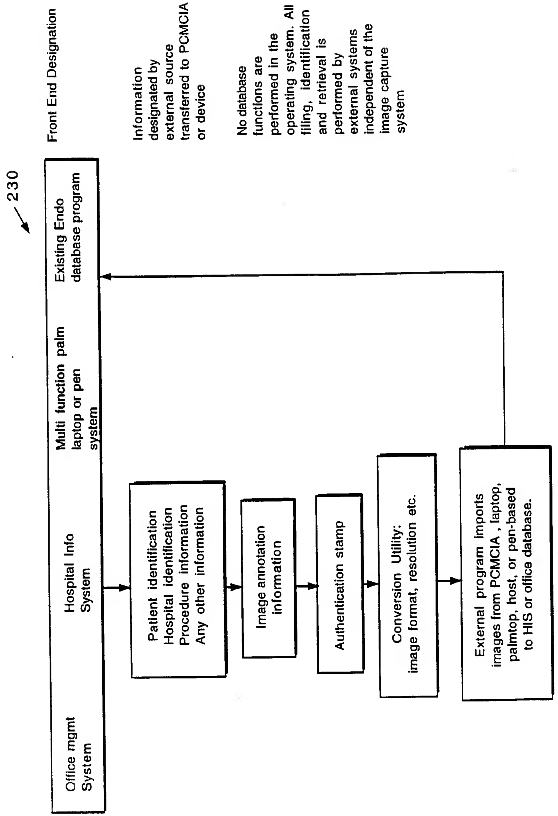

230
Office mgmt System
Hospital Info System
Multi function palm laptop or pen system
Existing Endo database program
Front End Designation
Information designated by external source transferred to PCMCIA or device
No database functions are performed in the operating system. All filing, identification and retrieval is performed by external systems independent of the image capture system
Patient identification
Hospital identification
Procedure information
Any other information
Image annotation information
Authentication stamp
Conversion Utility: image format, resolution etc.
External program imports images from PCMCIA , laptop, palmtop, host, or pen-based to HIS or office database.

FIG. 17

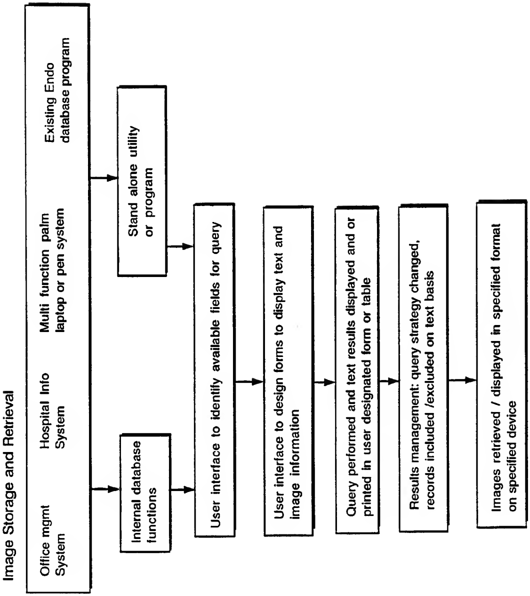


Image Storage and Retrieval
Office mgmt System
Hospital Info System
Multi function palm laptop or pen system
Existing Endo database program
Internal database functions
Stand alone utility or program
User interface to identify available fields for query
User interface to design forms to display text and image information
Query performed and text results displayed and or printed in user designated form or table
Results management: query strategy changed, records included /excluded on text basis
Images retrieved / displayed in specified format on specified device

FIG.18

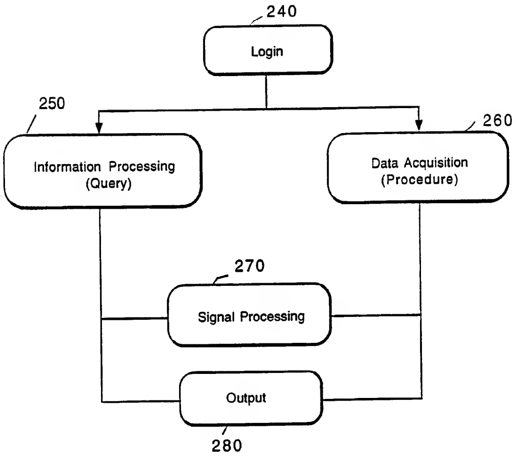

240
Login
250
Information Processing
(Query)
260
Data Acquisition
(Procedure)
270
Signal Processing
Output
280

FIG. 19

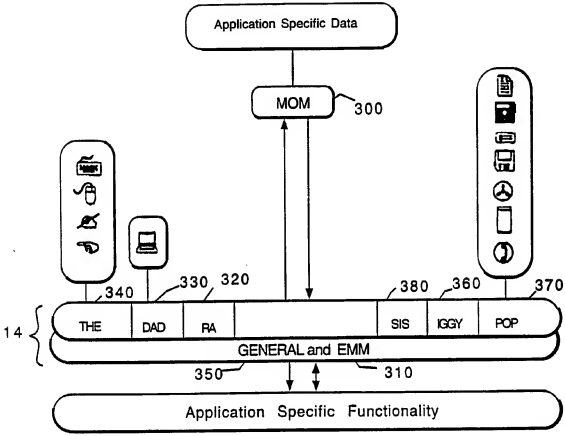
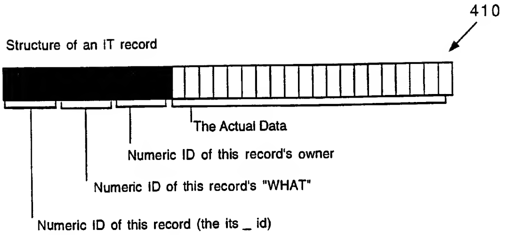
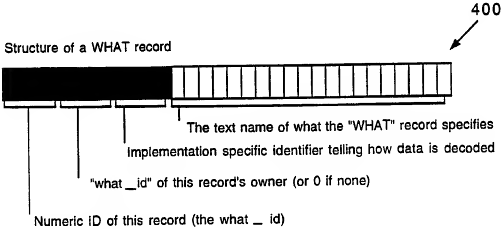

Application Specific Data
MOM
300
340
330
320
380
360
370
14
THE
DAD
RA
SIS
IGGY
POP
GENERAL and EMM
350
310
Application Specific Functionality

FIG. 20





<u>Sample Schema definition</u>

This sample is shown using RAIMA database language, but can be applied to any generic database handler.

```
    record what {                /*field types */
    unique key long what_id;     /*field key */
    key  long what_ owner;       /*key of its owner */
    char what_ name[32]          /*field name */
    long what_ look              /*it's appearance */
 }
```

FIG. 21

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/02919

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06F 15/00; A61N 1/30
US CL :364/413.13; 395/900; 604/20,21
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/413.13; 395/900; 604/20,21

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US, A, 4,817,050 (KOMATSU ET AL.) 28 MARCH 1989, SEE THE ENTIRE REFERENCE.	1
X --- Y	US, A, 5,049,147 (DANON) 17 SEPTEMBER 1991, SEE THE ENTIRE REFERENCE.	1 ------- 2-10

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 13 MAY 1994	Date of mailing of the international search report 04 AUG 1994
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer ROY ENVALL B. Nandell for
Facsimile No.	Telephone No. (703) 305-9711